LLNL-PROC-663432

Optimization of Plane Fits to Image Segments in Multi-View Stereo

N. Max, H. Kim

October 29, 2014

WACV 2015: IEEE Winter Conference on Applications of Computer Vision
Waikoloa Beach, HI, United States
January 6, 2015 through January 9, 2015

Optimization of Plane Fits to Image Segments in Multi-View Stereo

Nelson Max
University of California, Davis
max@cs.ucdavis.edu

Hyojin Kim
Lawrence Livermore National Laboratory
kim63@llnl.gov

Abstract

One step in the reconstruction pipeline for creating scene geometry from multiple camera views is the fitting of 3-D planes to image segments. We optimize the four homogeneous coefficients of the equation of a plane, so that the homography it induces on a source image segment minimizes a cost function with photoconsistency and edge consistency terms. The photoconsistency term for a segment is defined as the summed squared pixel color differences with all target images in which it is visible. For the edges we find a set of subpixel-precise Canny edge points from the source image. The edge consistency term considers all pairs of segments which are adjacent in 3D, and sums the squared 2D distances of the appropriate Canny edge points to the projection of the line in which the two segment planes intersect. We apply the L-BFGS non-linear optimization algorithm, which uses derivatives of the cost function with respect to the plane equation coefficients, and we derive analytic formulas for these derivatives for both terms in the cost function. An alternate optimization method uses the sparse Hessian matrix of second derivatives, and we compute these second derivatives as well. We test these plane optimizations on both synthetic and real image sequences.

1. Introduction

One goal in multi-image stereo analysis is to reconstruct a geometric model of a scene. In many cases, for example, in the urban scenes we investigate here, certain segments can be well approximated by planar geometry. The goal of this paper is to optimize the coefficients a, b, c, and d of the homogeneous plane equation

$$ax + by + cz + d = 0 \tag{1}$$

so that the homography it induces on a source image segment minimizes a cost function defined as the summed squared pixel color differences with all target images in which the segment is visible. To this we add an edge consistency term described in the abstract and in section 4.4.

Fua and Leclerc [6] optimize the geometry of a triangular mesh approximating an object by minimizing a cost that depends on the position coordinates of all of the vertices of the mesh, and on colors at a grid of texels on each triangle, compared with their projections with known camera parameters onto bilinearly interpolated target images. They use conjugate gradient minimization, and calculate the analytic derivatives of the cost function needed by this method. Inspired by that paper, we also use analytic derivatives of our cost function, using C^1 bicubic interpolation of the target image pixels, so that our cost function is a C^1 function of the plane equation coefficients.

Mitchell and Netravali [13] discuss the qualities of the members of the two parameter family of symmetric 4-segment C^1 piecewise cubic filter functions that form a "partition of unity" so that they exactly reconstruct constant functions, and we use these reconstruction filters here. We use the L-BFGS code of Bochkanov [2], which constructs an approximation of the Hessian second-derivative matrix for the cost function, using only evaluations of the cost and its first derivatives. This works well on functions that are only C^1 rather than C^2, but requires that they be smooth, which is why we do not use bilinear interpolation. Optionally, we also compute the second derivatives, for use with the STENMIN sparse-Hessian optimization method [3].

The space of 3D planes actually has only three degrees of freedom, since the equation can be put into a three parameter form by dividing by any coefficient which is non-zero, but the choice of which coefficient to divide by depends on the equation. To avoid considering separate cases, we have optimized over all four coefficients, and the fact that there is always a 4D direction along which the cost function is constant does not seem to cause problems in the optimization.

We work in the framework of Kim *et al.* [11]. We start with Mean Shift color segmentation (Comaniciu and Meer [5]) and exclude segments, based on their color and size, that are likely not to be planar building surfaces, but instead non-planar vegetation. We then initialize the plane equations for the remaining segments, and proceed with the non-linear optimization.

2. Related work

The proposed method requires initial plane coefficients, so another planar reconstruction algorithm is needed for initialization. Kim *et al.* [11] have presented such an algorithm to provide a set of planes for each segment, using a hierarchical search in the space of planes, parameterized by the locations where they intersect three viewing rays. However, it is a brute-force search by evaluating all possible planes that place the segment within the 3D bounding box, which requires a lot of computation time. Instead, their plane estimations at finer levels can be replaced with our plane optimization so that planes are quickly refined.

Another related method is Patch-Based Multi-View Stereo (PMVS) by Furukawa and Ponce [8]. This method gives a set of small patches (*e.g.* 5×5), each of which represents a plane. For each patch derived from a matched feature point, this method finds a plane that minimizes a cost function by rotating the plane normal around two axes.

Plane Sweeping, a similar planar reconstruction approach by Gallup *et al.* [9], also reconstructs a set of planes by using a set of initial feature correspondences and an intensity-based cost evaluation.

Furukawa *et al.* [7] selected planes that are perpendicular to one of the three axes of a coordinate system aligned to a “Manhattan world” urban scene, but we want to be able to handle slanted roofs and houses on curved suburban streets, so we optimize over the full three degrees of freedom for a plane, rather than the three-way choice plus one continuous degree of freedom valid for the restricted Manhattan world.

Zhang *et al.* [16] also use a cost function including color matching on image segments fit with planar surfaces, in their disparity initialization step. They use derivatives of this matching cost with respect to the plane equation coefficients for their non-linear optimization, but they estimate these derivatives by cubic Hermite interpolation between sampled finite differences, rather then by computing them analytically as we do.

Beniham and Malis [1] show how to approximate the Hessian matrix for a color matching term using only the first derivatives at the identity homography and at the actual homography. Our analytic Hessian also includes an edge consistency term that involves the planes of two adjacent segments, where this method may not apply.

3. Source segment regions

We currently take a segment from a single source view, and optimize its mapping onto multiple target views. But some segments may not be completely visible in all target views, either because they are mapped outside the target image frame, or due to occlusion. So we exclude target views in which the initial plane homography maps less than a threshold fraction t_1 of the segment pixels into the target image window. We consider source pixels to be occluded in a target view if their color difference with the interpolated target color at their mapped target position is greater than a threshold t_2, and similarly exclude target views where more than a fraction t_3 of the source segment pixels are occluded in this sense. Separately for each of the remaining n_t target views, we modify the source segment as follows.

We start with the subsegment of non-occluded source pixels, and dilate it with t_4 passes of adding 4-neighbor pixels, to close up small holes considered to be occluded by the above color difference test, which do not come from true occlusion. This will also add nearby pixels of contrasting colors from neighboring segments, assuring that the cost becomes positive for incorrect planes, even in the absence of texture on the source segment, but it can make the cost larger for the correct plane, due to cases in which the neighboring segments are at different depths. For example, the best match at a T-junction would be at an incorrect plane.

As in Fua and Leclerc [6], we assume that the initial plane homography maps source pixels to within a few image pixels of their correct positions, so that the local cost minimum found by the optimization is likely to be near the correct plane equation. Nevertheless, the adjustment of the plane could move some of the source pixels outside of the target image window. We use two methods to handle this possibility. First, we erode the source segment away from the inverse-mapped edges of the target image by t_5 erosion passes, to give a little wiggle room for the plane, within which the adjusted segment still maps inside the target image window. Second, we give large color values to the pixels outside the target image window when they are used in the piecewise bicubic interpolation. The effect that these large values have on the cost function and its derivatives drives the plane equation optimization to move the mapped pixels back inside the target image window if the plane moves outside of the wiggle room from the first method. But since we recompute the homographies and the eroded sets as we iteratively adjust the plane equations, the mapped segment is eventually allowed to move. In our current tests, we have used $t_1 = 0.5$, $t_2 = 20$ (out of 255), $t_3 = 0.24$, $t_4 = 2$, and $t_5 = 3$.

4. Cost function and its derivatives

4.1. Photoconsistency cost

For a source image segment S, and a candidate plane Q with equation (1) coefficients a, b, c, and d, our photoconsistency cost $C_{Sp}(a, b, c, d)$ is defined as follows.

Let $H_k(p)$ be the homography of a source image pixel p induced by the plane Q from segment S in the source image to the kth of the n_t target images in which it is considered visible, let Source(p) be the source image color at p, let S_k be the dilated and eroded segment constructed from S as above for target image k, and let $\tilde{f}_k(q)$ be the piecewise bicubic color interpolation at position q in target image k. Then our photoconsistency cost term for segment S is

$$C_{Sp}(a,b,c,d) = \sum_{k=1}^{n_t} \sum_{p \in S_k} |\, \tilde{f}_k(H_k(p)) - \text{Source}(p)|^2. \quad (2)$$

4.2. Derivatives of the homography

Let P_s be the 3 by 4 homogeneous matrix projecting from world coordinates $(x,y,z,1)^{\mathrm{T}}$ to homogeneous source image coordinates $(u',v',w')^{\mathrm{T}}$, so that the source image pixel coordinates are $(u,v,1)^{\mathrm{T}}$, with $u = u'/w'$ and $v = v'/w'$. Then we can insert an extra third row after the first two in this matrix, with entries from the plane equation coefficients (a,b,c,d) of plane Q, to give a matrix

$$\hat{P}_s = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ a & b & c & d \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (3)$$

so that $\hat{P}_s \cdot (x,y,z,1)^{\mathrm{T}} = (u',v',0,w')^{\mathrm{T}} \approx (u,v,0,1)^{\mathrm{T}}$ for any point (x,y,z) in plane Q. By adding this fourth row to make a square matrix, we can compute the inverse matrix $\hat{P}_s^{-1}$ whenever $\hat{P}_s$ is non-singular, which is whenever the plane Q does not pass through the viewpoint for the source image camera. Then $\hat{P}_s^{-1} \cdot (u,v,0,1)^{\mathrm{T}} \approx (x,y,z,1)^{\mathrm{T}}$ where (x,y,z) satisfies plane equation (1). If P_k is the 3 by 4 projection matrix for target image k, then

$$\hat{H} = P_k \hat{P}_s^{-1} \quad (4)$$

is a 3 by 4 matrix mapping $(u,v,0,1)^{\mathrm{T}}$ to a point $(r,s,1)^{\mathrm{T}}$ in the target image, constructed by projecting pixel (u,v) back along the source camera viewing ray onto the plane Q, and then along the target camera viewing ray onto target image k. Thus, if $\overline{\hat{P}_s^{-1}}$ is the 4 by 3 matrix formed from $\hat{P}_s^{-1}$ by removing its third column, then $H = P_k \overline{\hat{P}_s^{-1}}$ is the matrix for the homography induced by the plane Q.

To compute H, we use Cramer's rule for the matrix inverse

$$\hat{P}_s^{-1} = \hat{P}_s^{*} / |\hat{P}_s| \quad (5)$$

where $\hat{P}_s^{*}$ is the adjoint matrix of $\hat{P}_s$, the transpose of the matrix of minor determinants, whose (i,j)th entry is $(-1)^{i+j}$ times the determinant of the minor formed by removing the jth row and the ith column from $\hat{P}_s$, and $|\hat{P}_s|$ denotes the determinant of $\hat{P}_s$. Since H is a homogeneous matrix, it does not matter if all its entries are multiplied by a common factor, so we disregard the determinant $|\hat{P}_s|$ in the denominator of equation (5).

We are interested in the dependence of H on the plane equation coefficients a, b, c, and d. The minors for the first, second, and fourth columns of $\hat{P}_s^{*}$ all have one row of plane coefficients, so their determinants are linear expressions in a, b, c, and d, whose coefficients are the determinants of 2 by 2 minors of the 3 by 4 matrix P_s. Therefore, using equation (4), and equation (5) without its denominator, H can be written as a linear expression

$$H = H_a a + H_b b + H_c c + H_d d \quad (6)$$

where H_a, H_b, H_c, and H_d are 3 by 3 matrices whose entries are computed from those of P_s and P_k.

Computing the target image coordinates (r,s) for the pixel with indices (u,v) actually requires a division, since

$$H \cdot (u,v,1)^{\mathrm{T}} = (r',s',t')^{\mathrm{T}} \approx (r,s,1)^{\mathrm{T}} \quad (7)$$

with $r = r'/t'$ and $s = s'/t'$. Therefore we use the quotient rule to compute

$$\frac{\partial r}{\partial a} = \frac{\partial}{\partial a}\left(\frac{r'}{t'}\right) = \left(\frac{\partial r'}{\partial a} t' - \frac{\partial t'}{\partial a} r'\right) \Big/ t'^2 \quad (8)$$

using, for example, the first row of the matrix H_a to compute $\partial r'/\partial a$ for fixed u and v. Other partial derivatives of r and s with respect to a, b, c, and d are computed similarly.

4.3. Derivatives of the bicubic interpolation

As described in section 1 above, we use the two parameter family of symmetric 4-segment C^1 piecewise cubic filter functions $k(x)$ from equation (8) of Mitchell and Netravali [13]. We reparameterize these functions, which in [13] have support on the interval [-2, 2], to four cubic polynomials $g_i(x)$ on the interval [0, 1], for $i = 1,2,3,4$, so that when we reconstruct a function $f(r)$, the reconstructed function $\tilde{f}(r)$ for r in the interval $[j, j+1]$ is found from the sampled values $f(n)$ at integer positions n by the sum

$$\tilde{f}(r) = \sum_{i=1}^{4} f(j-2+i) g_i(x) = \quad (9)$$

$$(f(j-1), f(j), f(j+1), f(j+2)) \cdot M \cdot (x^3, x^2, x, 1)^{\mathrm{T}}$$

where $x = r - j$ and the 4 by 4 matrix M is given in terms of the B and C parameters of [13]. For the initial experiments described below, we have used $B = 0$ and $C = 0.5$, corresponding to the Catmull-Rom interpolating spline. By taking the derivatives $g_i'(x)$ of the polynomials $g_i(x)$ one can form a corresponding 4 by 3 matrix N with entries $n_{i,l} = (4-l) m_{i,l+1}$ so that the derivative $\tilde{f}'(r)$ is

$$\tilde{f}'(r) = \sum_{i=1}^{4} f(j-2+i) g_i'(x) =$$

$$(f(j-1), f(j), f(j+1), f(j+2)) \cdot N \cdot (x^2, x, 1)^{\mathrm{T}}. \quad (10)$$

For the piecewise bicubic reconstruction of $f(r,s)$ from its samples on the 2-D integer lattice, we use the tensor product to combine these filter functions in two dimensions, getting, for r in $[j, j+1]$, s in $[n, n+1]$, $x = r - j$ and $y = s - n$,

$$\tilde{f}(r,s) = \sum_{i=1}^{4}\sum_{m=1}^{4} f(j-2+i, n-2+m) g_i(x) g_m(y). \quad (11)$$

Taking the partial derivatives of this equation with respect to the plane equation coefficients, using the product rule and the chain rule, we get, for example,

$$\frac{\partial \tilde{f}(r,s)}{\partial a} = \sum_{i=1}^{4}\sum_{m=1}^{4} f(j-2+i, n-2+m) \left(g_i'(x) g_m(y) \frac{\partial r}{\partial a} + g_i(x) g_m'(y) \frac{\partial s}{\partial a} \right). \quad (12)$$

Using the power rule, we can now compute the partial derivatives of each term in the summations in equation (2). Thus for a grayscale image, letting $H_k(p) = (r, s)$,

$$\frac{\partial |\tilde{f}_k(H_k(p)) - \text{Source}(p)|^2}{\partial a} = 2\Big(\tilde{f}_k(H_k(p)) - \text{Source}(p)\Big) \frac{\partial \tilde{f}_k(r,s)}{\partial a}, \quad (13)$$

The partial derivatives with respect to the other coefficients are defined similarly.

Finally the partial derivatives of $C_{Sp}(a, b, c, d)$ in equation (2) can be found by summing the above equation over the n_t target images, over the pixels p in each modified segment segment S_k, and over the red, green, and blue color components if the images are in color.

The second partial derivatives are computed similarly. Note from equation (6) that H and therefore r', s', and t' are linear functions of the plane equation coefficients a, b, c, and d, so the factors $\frac{\partial r'}{\partial a}$ and $\frac{\partial t'}{\partial a}$ in equation (8) are constants independent of a, b, c, and d, but the factors r' and t' remain, so the quotient rule is still needed. The second partial derivatives of the tensor product splines can be calculated by the same method as for the first derivatives. The derivative computations are done in the loop that computes the color differences for the cost function itself, so they do not require extra memory accesses to the images.

4.4. Edge cost

For the edge consistency cost term, we first construct a set of edge points on the source image S, at zeros of the second directional derivative of the filtered image intensity in the gradient direction. These derivatives are computed at pixel centers by finite differences, as in [12], and then assumed to vary linearly along the horizontal and vertical lines connecting adjacent pixel centers, so that subpixel accurate edge points along these lines can be estimated. As in [4] and [12], we first select edge points whose edge strength (the negative of the interpolated finite difference estimates of the third derivative of the filtered image in the gradient direction) exceeds a strict threshold, shown as green dots in figure 1 (*top right*), and then extend the edges by "hysteresis" to adjacent zero crossings where the edge strength exceeds a looser threshold, shown as red dots in that figure. We call all these zero crossings Canny edge points.

For a pair of source image segments S and T which are adjacent in 2D, we find the set of pairs of adjacent pixels, one in S and one in T. Then we project these pixel pairs back into 3D onto their respective currently estimated segment planes Q_S and Q_T, and if the two 3D points are within a distance of t_6 times the distance between two diagonally adjacent pixels backprojected onto plane Q_S, we say that they are adjacent in 3D, and put them in a set $A(S,T)$, shown as the black dots in figure 1 (*bottom*). We say that the segments S and T are adjacent in 3D if they have at least t_7 such 3D adjacent pairs, and if the ratio of 3D adjacent pairs pairs to 2D adjacent pairs is above a threshold t_8. From this 3D adjacency criterion, we build an initial segment adjacency graph where there is an edge between nodes S and T if the two segments are adjacent in the above sense, and thus exclude segment pairs that are adjacent in 2D only across occlusion edges. We then modify this graph to eliminate adjacencies between pairs of segments whose planes are too close to parallel to give a reliable intersection line.

Now let S be a source segment, and T be another source segment that is connected to S by an edge in this 3D adjacency graph. We find the 3D intersection line L_{ST}^{3D} of the planes Q_S and Q_T of S and T, and its projection L_{ST} onto the plane of the source image, as shown in Figure 1 (*bottom*). We form a sub-collection K_{ST} of the Canny edge points which are within a Manhattan distance of t_9 of the set $A(S,T)$, and within a 2D distance t_{10} of the line L_{ST}, as shown as larger white dots surrounding some of the black dots in figure 1 (*bottom*). Note that for the vertical building corner occlusion edge between the pink selected vertical wall segment S and the adjacent green horizontal ground segment T, the white dots for K_{ST} do not extend to the top of this vertical edge, because of the distance condition to the green line L_{ST}. For the thresholds we currently use t_6 = 3, t_7 = 10, t_8 = .5, t_9 = 1, and t_{10} = 1.2.

We define an edge cost term for an edge line L_{ST} as

$$C_{STe} = \sum_{p \epsilon K_{ST}} d(p, L_{ST})^2 \quad (14)$$

where $d(p, L_{ST})$ is the 2D perpendicular distance of the point p to the line L_{ST}. Then the total edge cost is

$$C_e = \sum_{S,T \text{ adjacent}} C_{STe} \quad (15)$$

and the total cost to be optimized is

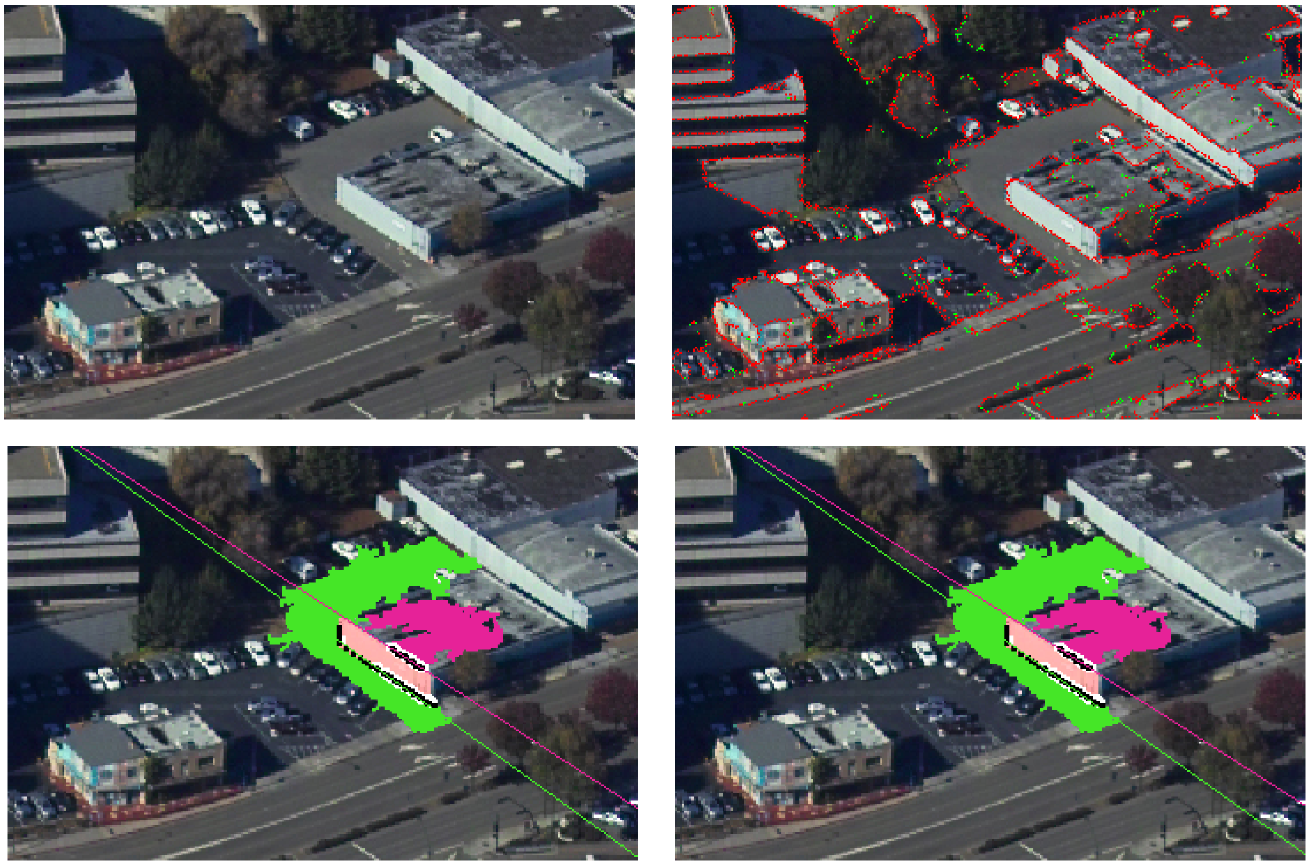

Figure 1. (*top-left*) The small boxed section of the original image of Walnut Creek at the top left of figure 2. (*top-right*) Same, with Canny edge points satisfying the strict threshold in green, and Canny edge points added by "hysteresis" in red. (*bottom-left*) A selected pink wall segment surrounded by black 3D adjacent point pairs, colored 3D adjacent roof and ground segments, and the projected lines in which their planes intersect the plane of the selected segment, shown in the adjacent segment color. (*bottom-right*) These projected intersection lines after the cost for the selected segment is optimized, keeping the other segments fixed.

$$C = \sum_S C_{Sp} + \alpha C_e \tag{16}$$

where α is a weight which we adjust to make the two terms have comparable values. For the scene in Figure 3, which has large segments, we used α = 50,000.

We now describe how to compute L_{ST} and the first and second derivatives of C_{ST} in terms of the coefficients a_S, b_S, c_S, and d_S, of Q_S and a_T, b_T, c_T, and d_T, of Q_T.

The family of planes (with coefficients a_{ST}, b_{ST}, c_{ST} and d_{ST}) passing through the 3D intersection line L^{3D}_{ST} is parameterized by a real parameter λ, with

$$a_{ST} = a_S + \lambda a_T \tag{17}$$
$$b_{ST} = b_S + \lambda b_T \tag{18}$$
$$c_{ST} = c_S + \lambda c_T \tag{19}$$
$$d_{ST} = d_S + \lambda d_T. \tag{20}$$

We solve for the parameter λ that makes this plane W_{ST} pass through the viewpoint $V = (v_x, v_y, v_z)$ of the source camera, so that $a_{ST}v_x + b_{ST}v_y + c_{ST}v_z + d_{ST} = 0$, giving an equation for λ:

$$\lambda = -\frac{a_S v_x + b_S v_y + c_S v_z + d_S}{a_T v_x + b_T v_y + c_T v_z + d_T}, \tag{21}$$

and therefore equations for a_{ST}, b_{ST}, c_{ST}, and d_{ST} in terms of a_S, b_S, c_S, d_S, a_T, b_T, c_T, and d_T.

The 2D projection L_{ST} of L^{3D}_{ST} onto the source image is the intersection of the plane W_{ST} and the image plane for the source camera. In section 4.2 we derived the matrix $\hat{P}_s$ which projects 3D points onto this image plane. Its third row involves a specific plane, but we can in fact use any plane here, since we only need the homogeneous coordinates u', v', and w' of $(u', v', q', w')^{\mathrm{T}}$ to compute the image pixel coordinates $(u, v, 1)^{\mathrm{T}}$, with $u = u'/w'$ and $v = v'/w'$. We therefore revised the third row of $\hat{P}_s$ to be simply (0, 0, 1, 0). By equation (5), the adjoint matrix $\hat{P}^*_s$ is proportional to the inverse of $\hat{P}_s$. According to the principles of linear algebra, the inverse transpose matrix $(\hat{P}^{-1}_s)^{\mathrm{T}} \approx (\hat{P}^*_s)^{\mathrm{T}}$ maps the equation for the plane W_{ST} to the equation for the corresponding plane in the camera viewing coordinates, so let

$$(r_1, r_2, r_3, r_4) = (a_{ST}, b_{ST}, c_{ST}, d_{ST})\hat{P}_s^* \qquad (22)$$

(transposing the equation to save space on the page). This plane passes through the origin of the viewing coordinates, so $r_4 = 0$, and its intersection with the image plane is the 2D projection L_{ST} of the 3D line L_{ST}^{3D}, with equation

$$r_1 x + r_2 y + r_3 = 0. \qquad (23)$$

Normalizing the line normal to a unit vector, we define

$$(s_1, s_2, s_3) = \frac{1}{\sqrt{r_1^2 + r_2^2}}(r_1, r_2, r_3). \qquad (24)$$

Then the squared distance of a Canny edge point $p = (x, y)$ to the line L_{ST} is

$$d(p, L_{ST})^2 = (s_1 x + s_2 y + s_3)^2. \qquad (25)$$

Thus the first and second derivatives of C_{STe} in equation (14) with respect to a_S, b_S, c_S, d_S, a_T, b_T, c_T, and d_T can be readily computed using the power, quotient, and chain rules of calculus. Note that the mixed partials like $\frac{\partial^2 C}{\partial a_S \partial c_T}$ are non-zero only if $S = T$ or the segments S and T are connected by an edge in the 3D adjacency graph, which is what makes the Hessian matrix sparse.

5. Experiments and discussion

Performance tests To evaluate the accuracy of our algorithm, we used a synthetic dataset from [10] with its ground-truth information, and simulated camera images rendered on an NVIDIA NVS 3100M graphics card using 16 times multisampling per pixel. We compared results from before and after applying our method in the hierarchical planar reconstruction framework of [11]. The photoconsistency-only matching results obtained from our optimization procedure when it is applied after the second pass are similar in accuracy to those obtained after four passes of the hierarchical reconstruction procedure. Compared with a single brute-force plane search through the four hierarchical levels, our method has a speedup of approximately 8 times on average.

Aerial and outdoor scene tests We also performed tests on several real datasets. For camera pose estimation, we used a structure-from-motion system called “Bundler” [14]. Figure 2 shows reconstruction results of an aerial dataset from [15]. Figure 3 shows reconstruction results of an outdoor dataset from [10]. To improve the reconstruction quality, we can iterate the optimization multiple times, as shown in the bottom of figure 3. As the iteration proceeds, planes are likely to converge to a global minimum of the cost function because each optimization process re-initializes the segment visibility. The L-BFGS and STENMIN non-linear optimization algorithms seem to perform equally well.

Acknowledgements This research was supported by grants DE-FG52-08NA28777 and DE-NA0000546 from the U.S. Department of Energy (DOE). This work was performed in part under the auspices of the U.S. DOE by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-PROC-663432.

References

[1] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 943–948, 2004. 2

[2] S. Bochkanov. www.alglib.net, 2007. 1

[3] A. Bouaricha. Algorithm 765: Stenmin: A software package for large, sparse unconstrained optimization using tensor methods. *ACM Transactions on Mathematical Software*, 23(1):81–90, 1997. 1

[4] J. Canny. A computational approach to edge detection. *IEEE-PAMI*, 8(6):679–698, 1986. 4

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE PAMI*, 24:603–619, 2002. 1

[6] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Juornal of Computer Vision*, 16:35–56, 1995. 1, 2

[7] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429, 2009. 2

[8] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE PAMI*, 32(8):1362–1376, 2009. 2

[9] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2

[10] H. Kim. Outdoor, aerial and synthetic datasets for multiview stereo. Website, 2011. available at `http://idav.ucdavis.edu/~hkim/mvs/dataset`. 6, 7

[11] H. Kim, Q. Hunter, M. Duchaineau, K. Joy, and N. Max. Gpu-friendly multi-view stereo for outdoor planar scene reconstruction. In *Eighth International Conference on Computer Vision Theory and Applications (VISAPP), part of VISIGRAPP*, pages 255–264, 2012. 1, 2, 6

[12] T. Lindenberg. Edge detection and ridge detection with automatic scale selection. In *Proceedings of CVPR-96*, 1996. 4

[13] D. Mitchell and A. Netravali. Reconstruction filters in computer graphics. *Computer Graphics*, 22(4), 1988. 1, 3

[14] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2), 2008. 6

[15] B. Wayne. Aerial images of Walnut Creek, California. Website, 2007. available at `http://www.cognigraph.com/walnut_creek_Nov_2005`. 6, 7

[16] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. Consistend depth map recovery from a video sequence. *IEEE-PAMI*, 31(6):974–988, 2009. 2

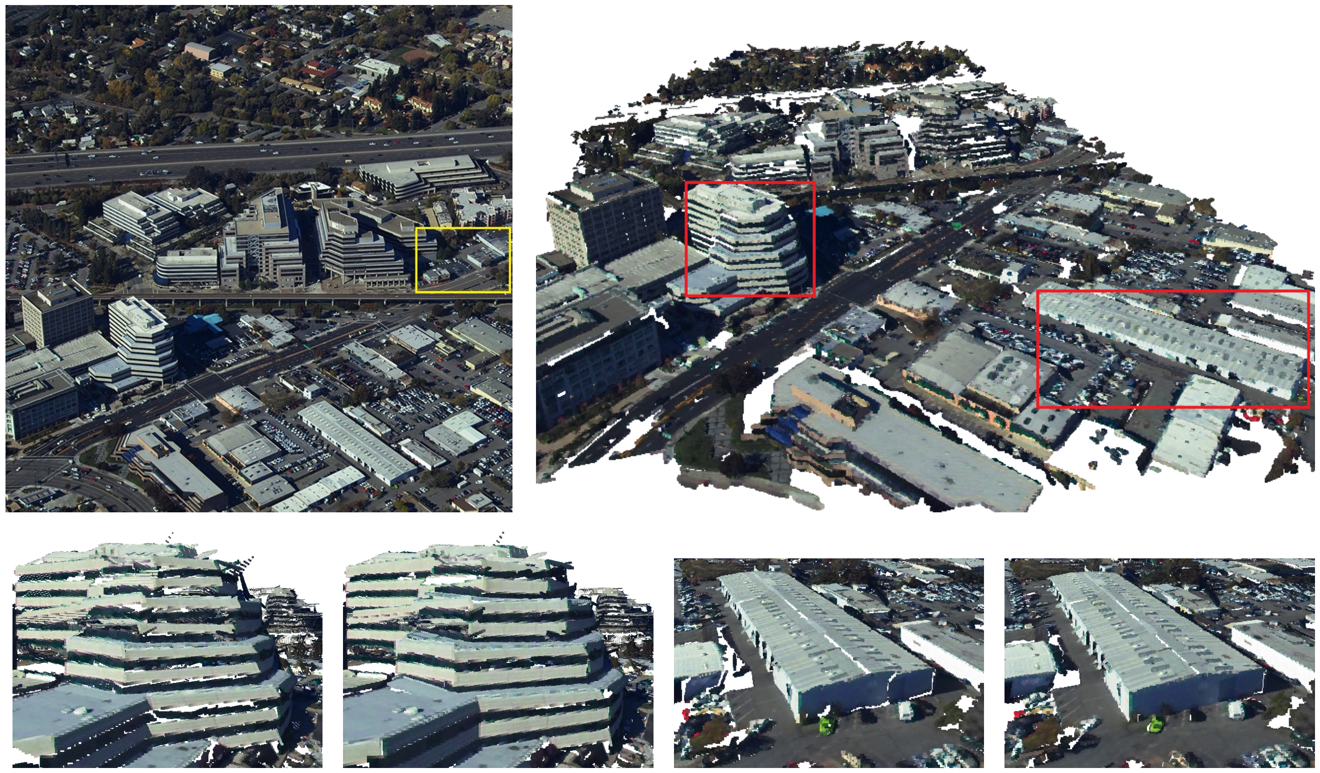
Figure 2. Reconstructed aerial urban scenes of Walnut Creek, CA, USA [15]. One of five original images (*top-left*), and a screen shot from the final reconstructed result (*top-right*). The bottom image pairs are results before and after applying our method, respectively.

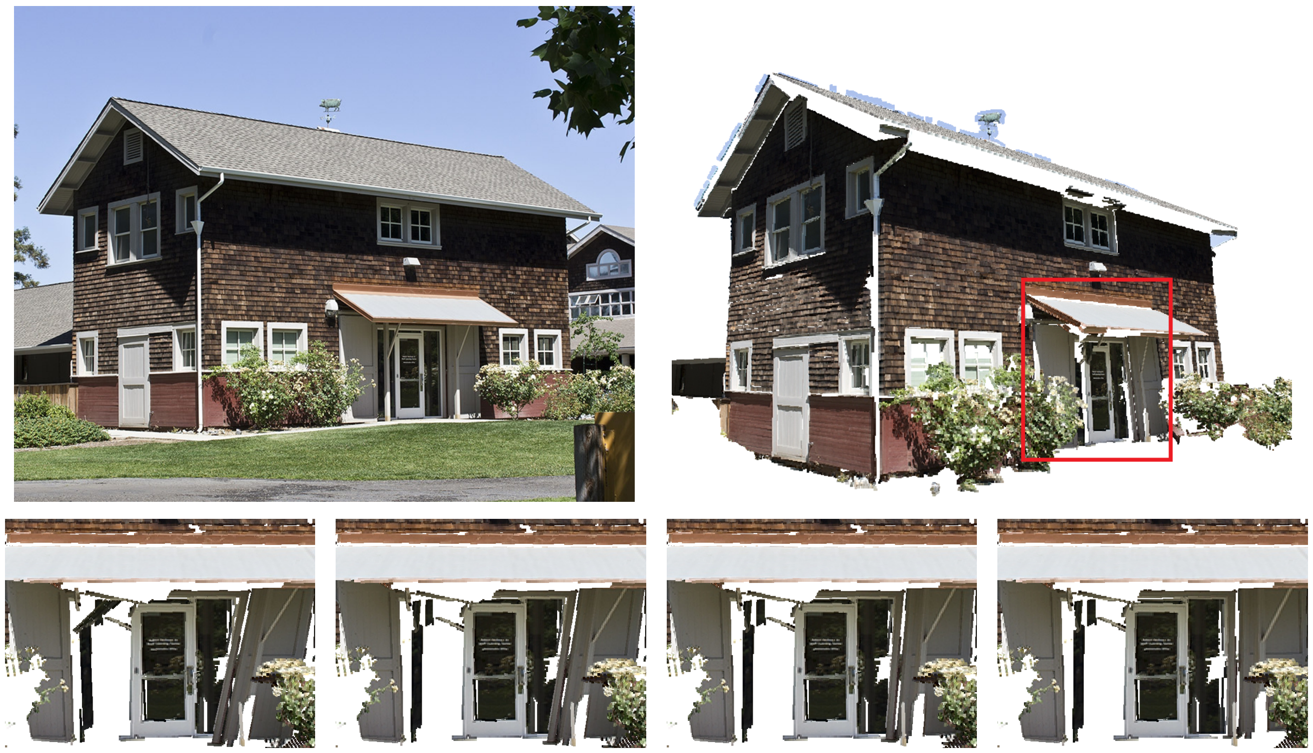
Figure 3. Reconstructed outdoor scenes from a dataset [10]. One of seven original images (*top-left*), and a screen shot from the final reconstructed result (*top-right*). The bottom images, from *left* to *right*, are results before our optimization, after optimizing 5 times, after optimizing 10 times, and optimizing of 20 times.